Project Report: Pathfinding for Emergency Services inHeidelberg, Germany

Link to GitHub Repository

GitHub link :https://github.com/Aaenoor/B208-Advanced-Algorithms

Objective

The main objective of this project is to apply and assess pathfinding algorithms in order to improve the effectiveness of emergency services in Heidelberg, Germany. The goal of this project is to employ Dijkstra's and A* algorithms to determine the shortest trip from any place in the city to the closest hospital.

Methodology

- **City Selection**
  - **Chosen City:** Heidelberg, Germany
  - **Reason:** Heidelberg is a mid-sized city, making it practical for detailed analysisand execution.
- **Data Collection**
  - **Geographic Information:** OpenStreetMap (OSM) was used to get geographic data on Heidelberg, including its topography, layout, and road network.
  - **Hospitals:** compiled a list of Heidelberg hospitals together with their coordinates for latitude and longitude

.

  - **Datasets Used:**
    - OSM data for the city's layout and road network
    - Hospital locations obtained from OSM features
  - **Dataset URL:** OpenStreetMap HYPERLINK "https://www.openstreetmap.org/" HYPERLINK "https://www.openstreetmap.org/"Data
- **Graph Mapping**

Nodes: Interest points, such as hospitals and other places.

Edges: Road network-based connections made between nodes

OSMnx: was the tool used to obtain and analyze the OSM data.

- **Algorithm Selection**

The well-known shortest path algorithm Dijkstra's Algorithm ensures that the shortest path in weighted graphs is found.

The A* Algorithm is a well-informed search algorithm that guarantees the shortest path while utilizing heuristics to increase search performance**.**

- **Theoretical Analysis**
    - **Effectiveness:**

It is anticipated that both algorithms will correctly determine the quickest route to the closest hospital.

A*'s heuristic-based methodology is anticipated to yield somewhat better results.

-
    - **Efficiency:**
        - **Dijkstra's Algorithm:**
            - O(V2) is the time complexity.O(V^2)For dense graphs, the answer is O(V2); for graphs with a priority queue, it is O((V+E)logV)O((V + E) \log V)O((V+E)logV)..
            - Space Complexity: O(V)O(V)O(V).
        - **A* Algorithm:**
            - Complexity of Time: O((V+E)logV)Similar to Dijkstra's, but frequently quicker in reality because of heuristics, is O((V + E) \log V)O((V+E)logV..
            - Space Complexity: O(V)O(V)O(V).
- **Implementation**
    - Implemented using Python with libraries such as NetworkX and OSMnx.
    - **Scripts:**
        - `data_collection.py`: Collects and processes data.
        - `graph_mapping.py`: Maps the city graph and adds hospitals.
        - `pathfinding.py`: Contains the pathfinding logic and user interaction.
        - `empirical_evaluation.py`: Evaluates the algorithms using predefinedscenarios.
- **Empirical Evaluation**
    - **Scenarios:**
        - **Neuenheim (North Heidelberg)**
        - **Heidelberg Zoo (West Heidelberg)**
        - **Heidelberg-Süd (South Heidelberg)**
        - **Handschuhsheim (North-East Heidelberg)**
        - **Boxberg (South-West Heidelberg)**
    - **Metrics:**
        - **Runtime:** Time taken to find the path.
        - **Memory Usage:** Memory used during the computation.

    - **Path Accuracy:** Comparison with best-known routes.
- **Results:**
    - Dijkstra and A* both consistently found the shortest paths with slightperformance differences.
    - A* generally performed faster than Dijkstra.
    - Memory usage was similar for both algorithms.

Empirical Evaluation Results

Scenario: Neuenheim (North Heidelberg)

- **Nearest Hospital:** Klinik für Allgemeine Innere Medizin und Psychosomatik
- **Runtime:** 0.36 seconds
- **Memory Usage:** 0.77 MB
- **Dijkstra's Path:** Max-Reger-Straße → Quinckestraße → Seitzstraße → Gundolfstraße → Mönchhofstraße → Unnamed Road → Im Neuenheimer Feld → Kirschnerstraße → Hofmeisterweg → Kirschnerstraße.

- **A* Path***:* Max-Reger-Straße → Quinckestraße → Seitzstraße → Gundolfstraße → Mönchhofstraße → Unnamed Road → Im Neuenheimer Feld → Kirschnerstraße →Hofmeisterweg → Kirschnerstraße.

Scenario: Heidelberg Zoo (West Heidelberg)

- **Nearest Hospital:** Institut für Humangenetik
- **Runtime:** 0.28 seconds
- **Memory Usage:** 0.72 MB
- **Dijkstra's Path:** Im Neuenheimer Feld.
- **A* Path:** Im Neuenheimer Feld.

Scenario: Heidelberg-Süd (South Heidelberg)

- **Nearest Hospital:** Kurpfalzkrankenhaus
- **Runtime:** 0.40 seconds
- **Memory Usage:** 0.72 MB
- **Dijkstra's Path:** Czernyring → Bergheimer Straße → Mannheimer Straße → Ludwig-Guttmann-Straße.
- **A* Path:** Czernyring → Bergheimer Straße → Mannheimer Straße → Ludwig-Guttmann-Straße.

Scenario: Handschuhsheim (North-East Heidelberg)

- **Nearest Hospital:** Institut für Humangenetik
- **Runtime:** 0.41 seconds

- **Memory Usage:** 0.72 MB
- **Dijkstra's Path:** Handschuhsheimer Landstraße → Pfarrgasse → Zeppelinstraße →Berliner Straße → Im Neuenheimer Feld.
- **A* Path:** Handschuhsheimer Landstraße → Pfarrgasse → Zeppelinstraße → BerlinerStraße → Im Neuenheimer Feld.

Scenario: Boxberg (South-West Heidelberg)

- **Nearest Hospital:** Klinik für Allgemeine Innere Medizin und Psychosomatik
- **Runtime:** 0.51 seconds
- **Memory Usage:** 0.73 MB
- **Dijkstra's Path:** Im Beind → Leimer Straße → Rathausstraße → Herrenwiesenstraße →Am Rohrbach → Römerstraße → Lessingstraße → Mittermaierstraße → Ernst-Walz-Brücke → Berliner Straße → Mittermaierstraße → Berliner Straße → Jahnstraße → Im Neuenheimer Feld → Kirschnerstraße → Hofmeisterweg → Kirschnerstraße.
- **A* Path:** Im Beind → Leimer Straße → Rathausstraße → Herrenwiesenstraße → Am Rohrbach → Römerstraße → Lessingstraße → Mittermaierstraße → Ernst-Walz-Brücke
→ Berliner Straße → Mittermaierstraße → Berliner Straße → Jahnstraße → Im Neuenheimer Feld → Kirschnerstraße → Hofmeisterweg → Kirschnerstraße.

Design Decisions

- **City Selection:** Heidelberg was selected as the city because of its data availability and reasonable size.
- **Data Collection:** OSM data was used to ensure correctness and comprehensiveness..
- **Graph Mapping:** Ensured a realistic representation of Heidelberg's road network.
- **Algorithm Selection:** Dijkstra for guaranteed shortest path; A* for heuristicoptimization.
- **Empirical Scenarios:** Selected diverse locations to validate the algorithmscomprehensively.

Conclusion

- Both Dijkstra's and A* algorithms are effective in finding the shortest paths to the nearesthospitals in Heidelberg.
- A* generally performed faster with similar memory usage, making it a more efficientchoice in practice.

Dataset URL

- OpenStreetMap Data